Defence Research and Development Canada
Recherche et développement pour la défense Canada

Engineering Review of ANCÆUS/AVATAR

An Enabling Technology for the Autonomous Land Systems Program?

G. Broten, D. Erickson, J. Giesbrecht, S. Monckton, S. Verret
Defence R&D Canada – Suffield

Technical Report
DRDC Suffield TR 2003-167
December 2003

Canada

Engineering Review of ANCÆUS/AVATAR

An Enabling Technology for the Autonomous Land Systems Program?

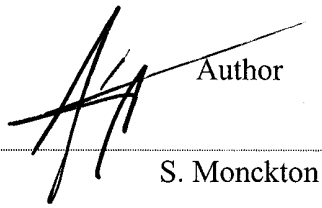
G. Broten, D. Erickson, J. Giesbrecht, S. Monckton, S. Verret
Defence R&D Canada – Suffield

Defence R&D Canada – Suffield

Technical Report

DRDC Suffield TR 2003-167

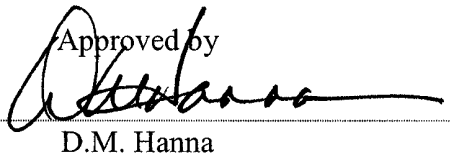
December 2003



Author

S. Monckton

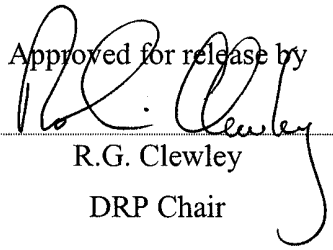
Approved by



D.M. Hanna

HTVSS

Approved for release by



R.G. Clewley

DRP Chair

Abstract

Following the guidelines of the Engineering Review Terms of Reference, this report examines whether ANCÆUS and/or VCS (Vehicle Control Station) constitute an enabling technology for future Autonomous Land System (ALS) autonomous vehicles. Since 1989, forward thinking engineering has characterized the history of ANCÆUS and VCS. ANCÆUS has broken ground in network vehicle architectures, vehicle modularity, hardware self-discovery and robust multi-vehicle networking. Similarly, VCS has grown into a major business for CDL Systems Inc. through consistent technical superiority. Though conceived over 15 years ago, these systems remain sufficiently relevant and desirable to serve within the ILDS deployed to Afghanistan in 2003. This report reviews the historical background and current technical content of ANCÆUS, VCS, ALS and similar commercial/open source efforts. The subsequent comparative evaluation of ANCÆUS and commercial/open source technologies revealed that currently no single system constitutes a complete a multi-robot autonomous system architecture. The report concluded that while ANCÆUS was found to be technically well conceived, and often well executed, commercial/open source technologies are more capable in a multi-vehicle context and represent a lower technical risk. Specifically, the report concludes:

1. ANCÆUS is not an enabling technology for future autonomous vehicles and that

2. ALS should adopt commercial/open source technology to support a new ALS architecture and

3. ALS should provide ANCÆUS compatibility and evaluate the ANCAEUS control codes as a guideline for future ALS vehicle interfaces.

Résumé

Conformément aux lignes directrices des paramètres de la Revue technique, ce rapport examine si ANCÆUS et / ou la SCV (Station de contrôle de véhicules) constituent une technologie de mise en service des Systèmes terrestres autonomes (STA) futurs pour les véhicules autonomes. Depuis 1989, une ingénierie de réflexion prospective a caractérisé l'histoire d'ANCÆUS et SCV. ANCÆUS a innové en matière d'architecture des véhicules réseautés, de modularité des véhicules, d'auto-découverte de matériel de traitement des données et de réseautage robuste de véhicules groupés. De même, de par sa supériorité technique constante, la SCV a donné le jour à un chiffre d'affaire important pour CDL Systems Inc. Bien qu'ils aient été conçus il y a plus de 15 ans, ces systèmes demeurent suffisamment applicables et désirables pour être utilisés dans les Systèmes améliorés de détection de mines terrestres (ILDS) déployés en Afghanistan, en 2003. Ce rapport étudie le contexte historique et le contenu technique actuel des ANCÆUS, SCV, STA ainsi que des efforts similaires commerciaux ou en source libre. L'évaluation comparative ultérieure des ANCÆUS et des technologies commercialisées / de sources libres a révélé qu'actuellement aucun système unique ne constitue une architecture complète de système autonome de robots multiples. Le rapport conclut que bien que ANCÆUS soit techniquement mieux conçu et souvent bien exécuté, les technologies commerciales / de sources libres sont plus performantes dans un contexte de véhicules multiples et qu'elles présentent un risque technique moindre. Le rapport conclut spécifiquement que :

1. ANCÆUS n'est pas une technologie de mise en service des véhicules autonomes du futur ;

2. les STA devraient adopter une technologie de commercialisation / de sources libres pour soutenir une nouvelle architecture des STA ;

3. les STA devraient être compatibles avec ANCÆUS et devraient évaluer les codes de contrôle d'ANCAEUS comme ligne directrice pour les interfaces futures de véhicules STA.

Executive summary

This report attempts to determine whether ANCÆUS v3.0, a teleoperation platform developed since 1989, constitutes an enabling technology for future Autonomous Land System (ALS) autonomous vehicles. To achieve this end, the historical background and current technical content of ANCÆUS, VCS, ALS and commercial/open source programs were reviewed, a variety of technical meetings and discussions were conducted and a group consensus documented. The team concluded that

- ANCÆUS is *not* an enabling technology for future autonomous vehicles and that
- ALS should adopt commercial/open source technology to support a new ALS architecture and
- ALS should provide ANCÆUS compatibility and evaluate the ANCÆUS control codes as a guideline for future ALS vehicle interfaces.

Following the guidelines of the *Engineering Review Terms of Reference*,the background of the ANCÆUS /Avatar, VCS and the ALS programs were examined to understand each group's technical objectives. Since 1989, the history of ANCÆUS and VCS have been characterized by forward thinking engineering. In particular, ANCÆUS has broken ground in network vehicle architectures, vehicle modularity, hardware self-discovery and robust multivehicle networking. Similarly VCS has grown into a major business for CDL Systems through consistent technical superiority. Though conceived over 15 years ago, each of these characteristics is remains sufficiently relevant and desirable today to serve within the ILDS currently deployed in Afghanistan.

A technical audit examined the current state of software, hardware and documentation of both ANCÆUS /Avatar system and VCS in the context of provisional design requirements for a multi-robot autonomous systems (MRAS). For further context, additional technical audits were performed on alternative commercial/open source systems such as Saphira/OAA, Player/Stage, CLARAty, and CARMEN. Design requirements for MRAS were assembled to measure the viability of these systems as enabling technologies for the Autonomous Land Systems Program. The subsequent evaluation of ANCÆUS and commercial/open source technologies revealed that none form a complete architecture for MRAS. Indeed, set in the context of Breummer's levels of autonomy, no system appears to meet the needs of full autonomy. However, commercial/open source autonomous operating systems appear to have greater autonomous capability in comparison to either ANCÆUS or VCS teleoperation systems. Not surprisingly, as the target 'level of autonomy' increases the suitability of any candidate appears to decrease. In general ANCÆUS was found to be technically well conceived, and often well executed though using somewhat dated methods. Significantly, ANCÆUS is very poorly documented, the most recent formal documentation (v2.1) is two years old, exacerbated by source code documentation well below industry standard.

A further comparison studied the future effort required to meet the needs of full autonomy for these systems. This subjective study revealed that, regardless of candidate, there would be significant effort to improve ANCÆUS and commercial/open source systems and that teleoperation represents a relatively small, though necessary, portion of any MRAS development. As implied by the technical audit, the level of effort needed to expand ANCÆUS or the other alternative technologies increased with the level of autonomy. Significantly, greater effort would be needed to expand ANCÆUS or VCS to meet full autonomy than to extend alternative technologies. These difficulties become more pronounced in the case of VCS, given CDL Systems' significant intellectual holdings in the software.

Based on the technical, scientific, and business aspects involved the following options were explored:

1. Adopt ANCÆUS to control all low-level ALS platforms and then develop/adopt a high-level software technology to work above ANCÆUS for full autonomy.

2. Overhaul the ANCÆUS system to include high-level software technology to be capable of full autonomy.

3. Adopt/develop an alternative software technology using the ANCÆUS control codes for all ALS platforms.

4. Adopt/develop an alternative software technology but maintain a low-level ANCÆUS API for legacy ANCÆUS platforms.

5. Adopt an existing alternative software technology and hardware standards to support a new ALS architecture.

6. Build a new ALS software technology according to a new design philosophy from the ground up.

These options represent a sliding scale of commitment to ANCÆUS technology. The consensus of the team lies between option 3 and 5. In general the group feels the ANCÆUS technology was a remarkable achievement by a small staff using mid-90's technical standards. The group feels that there is no reason ANCÆUS support could not be provided by future ALS work and strongly recommends this should be done. However, the group felt that for a variety of technical reasons, the adoption of ANCÆUS without modification (i.e. option 1) would not be acceptable and that a major overhaul (option 2) would be costly and redundant with other commercial and open source systems. Nevertheless, the ANCÆUS project has, over time, developed a consistent and useful command set that could easily be adopted for future ALS equipment. At the very least, the team recommends supporting ANCÆUS through option 4, perhaps using option 3 if the API proves sufficiently flexible.

G. Broten, D. Erickson, J. Giesbrecht, S. Monckton, S. Verret. 2003. Engineering Review of ANÆUS/AVATAR. DRDC Suffield TR 2003-167. Defence R&D Canada – Suffield.

Sommaire

Ce rapport tente de déterminer si l'ANCÆUS v3.0, une plate-forme de téléopération mise au point depuis 1989, constitue une technologie de mise en service pour les véhicules autonomes de systèmes terrestres autonomes futurs. Pour atteindre cet objectif, le contexte historique et le contenu technique actuel des ANCÆUS, SCV, STA ainsi que les programmes de commercialisation / de sources libres ont été examinées. Une variété de réunions techniques et de discussions on été conduites et un consensus de groupe a été documenté. L'équipe a conclu que :

- ANCÆUS n'est pas une technologie de mise en service des véhicules autonomes du futur ;

- les STA devraient adopter une technologie de commercialisation / de sources libres pour soutenir une nouvelle architecture de STA ;

- les STA devraient être compatibles avec ANCÆUS et devraient évaluer les codes de contrôle d'ANCAEUS comme ligne directrice pour les interfaces futures de véhicules STA.

Conformément aux lignes directrices des paramètres de la Revue technique, ce rapport examine le contexte d'ANCÆUS et / ou la SCV et les programmes de STA ont été examinés pour comprendre tous les objectifs techniques de chacun des groupes. Depuis 1989, l'histoire d'ANCÆUS et de la SCV a été caractérisée par une ingénierie de réflexion prospective. ANCÆUS, en particulier, a innové en matière d'architecture de véhicules réseautés, de modularité des véhicules, d'auto-découverte de matériel de traitement des données et de réseautage robuste de véhicules groupés. De même, la SCV, de par sa supériorité technique constante, a donné le jour à un chiffre d'affaires important pour CDL Systems Inc.. Bien qu'ils aient été conçus il y a plus de 15 ans, ces systèmes demeurent suffisamment applicables et désirables pour être utilisés dans les Systèmes améliorés de détection de mines terrestres (ILDS) déployés actuellement en Afghanistan.

Une vérification technique a examiné l'état actuel des logiciels, du matériel et de la documentation du système ANCÆUS /Avatar et de la SCV, dans le contexte des caractéristiques exigées provisoires pour des systèmes autonomes de robots multiples (SARM). Pour un contexte plus approfondi, on a exécuté des vérifications techniques additionnelles au sujet de systèmes de rechange de commercialisation / de sources libres tels que Saphira/OAA, Player/Stage, CLARAty, et CARMEN. Les exigences de conception pour les SARM ont été assemblées pour mesurer la viabilité de ces systèmes comme technologies de mise en service pour le Programme de Systèmes terrestres autonomes. L'évaluation comparative ultérieure d'ANCÆUS et des technologies commercialisées / de sources libres a révélé qu'aucun ne forme une architecture complète de système autonome de SARM. En fait, en se basant sur le contexte des niveaux d'autonomie de Breummer, aucun système ne semble répondre aux besoins d'autonomie complète. Les systèmes d'opération autonomes de commercialisation / de sources libres, semblent posséder une meilleure capacité d'autonomie, si on les compare soit au système de téléopération ANCÆUS soit à ceux de la SCV. Il n'est

pas surprenant que l'augmentation des « niveaux d'autonomie » des cibles aient pour résultat de diminuer la capacité d'adaptation des technologies candidates. En général, on trouve que ANCÆUS est bien conçu techniquement et qu'il est souvent bien exécuté bien qu'il utilise des méthodes moins modernes. De manière significative, ANCÆUS est très mal documenté, et la documentation officielle la plus récente (v2.1) date de deux ans sans compter que la documentation du code source est bien au-dessous de la norme de l'industrie.

Une comparaison plus approfondie a étudié les efforts qui seront requis à l'avenir pour répondre aux besoins d'autonomie complète de ces systèmes. Cette étude subjective révèle que sans tenir compte des technologies candidates, il faudrait un effort soutenu pour améliorer ANCÆUS et les systèmes de commercialisation / de sources libres ; bien qu'elle soit nécessaire, la télé-opération ne représente qu'une portion relativement petite de la mise au point des SARM. Tel que laissé entendre par la vérification technique, le niveau d'effort requis pour développer ANCÆUS ou les technologies de rechange s'est accru avec le niveau d'autonomie. De manière signifiante, développer ANCÆUS ou la SCV requiert un effort plus soutenu que de développer les technologies de rechange. Ces difficultés deviennent d'autant plus prononcées dans le cas de la SCV puisque CDL Systems a un investissement intellectuel considérable au niveau des logiciels.

En se fondant sur les aspects techniques, scientifiques et commerciaux existants, les options suivantes ont été explorées :

1. Adopter ANCÆUS pour contrôler les plates-formes de bas niveau, puis mettre au point ou adopter une technologie de logiciels de haut niveau pour travailler au-dessus d'ANCÆUS en visant une autonomie complète.

2. Réviser le système ANCÆUS pour y inclure une technologie de logiciels de haut niveau visant une capacité d'autonomie complète.

3. Adopter / mettre au point une technologie de logiciels de rechange en utilisant les codes de contrôle d'ANCÆUS pour les plates-formes de STA.

4. Adopter / mettre au point une technologie de logiciels de rechange mais maintenir un bas niveau d'interface du programme d'application (API) d'ANCÆUS comme ancienne application des plates-formes ANCÆUS.

5. Adopter une technologie existante de logiciels de solution de rechange ainsi que des normes de matériel pour soutenir une nouvelle architecture de STA.

6. Créer une nouvelle technologie de logiciels de STA conformes à la nouvelle philosophie de concepts, à partir du tout début.

Ces options représentent une échelle mobile en termes d'engagement à la technologie d'ANCÆUS. Le consensus de cette équipe repose entre l'option 3 et 5. En général, le groupe estime que la technologie d'ANCÆUS a produit des résultats remarquables du fait qu'il s'agissait d'une petite équipe de personnel utilisant les normes techniques des années 1990. Le groupe estime qu'il n'y a pas de raison de ne pas soutenir ANCÆUS pour de futurs travaux avec les STA et ce soutien est fortement recommandé. Le groupe estime cependant

que pour un certain nombre de raisons techniques, adopter ANCÆUS sans le modifier (option 1) ne serait pas satisfaisant et qu'une révision majeure (option 2) serait coûteuse et redondante par rapport aux autres systèmes de commercialisation et de sources libres. Le projet ANCÆUS a pourtant mis au point au cours du temps un jeu de commande conséquent et utile qui pourrait être facilement adopté à l'avenir pour l'équipement des STA. L'équipe recommande certainement de soutenir ANCÆUS au moyen de la quatrième option, peut-être au moyen de la troisième option si l'interface du programme d'application (API) se montre suffisamment adaptable.

G. Broten, D. Erickson, J. Giesbrecht, S. Monckton, S. Verret. 2003. Engineering Review of ANÆUS/AVATAR. DRDC Suffield TR 2003-167. R & D pour la défense Canada – Suffield.

This page intentionally left blank.

Table of contents

List of figures

List of tables

This page intentionally left blank.

1. Background

The growth of any technology program is the motivated by both technical vision and percieved needs. A brief historical overview of the elements involved in TVSS program sheds light on the motivations that have evolved into ANCÆUS/AVATAR, VCS, and the ALS program.

1.1 Autonomous Land Systems (ALS) Program

Defence Research and Development Canada (DRDC), on behalf of Canadian Department of National Defence, is developing new and innovative land vehicles.

In 1999 DRDC conducted a technology review that identified 21 key areas that will be the focus of DRDC's future R and D investments. Autonomous Intelligent Systems (AIS) was identified as one of the 21 key technologies by this review. Autonomous Intelligent Systems are defined as "automated or robotic systems that operate and interact in the complex and unstructured environments of the future battlespace"[TIS]. The AIS program has a mandate to conduct Research and Development to determine the capabilities of autonomous systems in "performing complex tasks through the perception and understanding of unstructured environments with minimal human direction or oversight"[TIS]. The AIS research is has three defined theatres of operation: Unmanned Air (UAV), Ground (UGV) and Underwater (UUV).

The portion of the AIS research that is dedicated to the ground theatre of operations is know the the Autonomous Land Systems (ALS) program. The ALS research is being conducted by the Tactical Vehicles Systems Section (TVSS) at DRDC Suffield. TVSS has a long history of developing unmanned ground vehicles (UGVs) that are controlled via tele-operation. Under the ALS program TVSS will research and develop autonomous UGVs that are capable of operating in both indoor and outdoor environments. The autonomous capabilities of these vehicles will allow them to navigate unstructured environments without the reliance on a human operator.

The ALS research is expected to define the future capabilities of UGVs for the time-frame of 2010 to 2020.

1.2 ANCÆUS/AVATAR

ANCÆUS/AVATAR technology, born from of an effort to develop remote controlled target tow vehicles in the late 1980's, has grown into a system capable of multivehicle remote control and limited autonomous capability. In 1989, DRES developed and fielded a remote controlled target system based on commercial R/C technology.

While the trial was successful, it was equally clear that the concept of remote controlled targets deserved a renewed engineering effort. In particular, the development group felt that a 'command-response' architecture was necessary to provide operators

with the confidence that the system was performing as directed. The ANCÆUS concept revolved around four basic elements:

- a vehicle
- a vehicle 'personality module' (VPM)
- a vehicle command processor (VCP)
- a control station

The control station would broadcast a command set over an RF modem. The VCP would interpret the command and send an appropriate acknowledgement to the host. The VCP would pass the translated commands to the VPM that would distribute appropriate commands to the devices mounted on the specific vehicle. This architecture encapsulated vehicle idiosyncracies into the VPM, permitting the VCP and host to remain unchanged from vehicle to vehicle.

The first fielded result appeared in 1990 in a 'tow missile shoot'. This system, deployed on an ODG Argo platform, employed a low bandwidth (2400 baud) data link and FM video link. Subsequent target trials built on this success, including the Argo based 1991 ERYX missile shoot in 1991.

By 1993 the focus had shifted from target towing to demining. The intellectual property surrounding the basic target tow vehicle (but not ANCÆUS) had passed largely to Boeing's target drone division (later sold to Bristol Aerospace and then Schreiner). The application area had shifted from targets to demining. Notably, ANCÆUS was used for the three 8×8 Argo ROMMIDS vehicles prepared (but never deployed) for Somalia.

The growing potential of the ANCÆUS system prompted DRES to launch a commercialization study and third party vehicle conversion in 1995 [6] (Alberta Research Council, Project Managers; Robotech Industries, vehicle subcontractors).The underlying technology had evolved into an Amiga 3000 or 2000 computer (MC68030) linked over RF 9600 baud modems to the platform vehicles. Though ANCAEUS could support up to 255 vehicles, RF modem problems limited the number of vehicles under simultaneous control to three.

The report reviewed DRES supplied source code, documentation, and references and discussed a D6 bulldozer ANCÆUS conversion by Robotech Industries. The report concluded that while the system had demonstrated successful application and was, in principal, attractive to potential industrial clients, the technology appeared to be poorly documented, practically difficult for third party vendors to adopt, and, in places, overly committed to hand coded assembly language modules. Further, the report warned that DRES' proposed adoption of OS/2 would hinder commercialization, given Windows95/NT growing market and 'mind' share amongst technical users at that time.

In 1997, the program focus on demining led to the successful CDN$6M ILDP technology demonstration program. The ILDP vehicle carried a suite of sensors

(vision, I/R, and TNA) that could be used to find and identify potential landmines. This technology was passed to General Dynamics to become the ILDS vehicle.

Entering its second decade, ANCÆUS shifted focus again towards combined operations. Urban Ram in 2001 demonstrated the first 'mixed operations' application of ANCÆUS technology with the use of a ram mounted D6 Cat moving with and supporting urban infantry operations. This mixed operational role is now exemplified in the (Multiagent Testing System), a CBN detection array mounted on a revised ANCÆUS platform known as ANCÆUS/AVATAR .

The ANCÆUS/AVATAR system is currently fielded within the ILDS system in Afghanistan and will be fielded again within the MATS system now under construction.

1.3 Vehicle Control Station (VCS)

The VCS system is a general purpose software system for controlling a variety of land, sea and air vehicles, and their payloads [3]. The VCS can be used to control unmanned air, water and ground vehicles for land mine detection, remote surveillance, reconnaissance, and target applications [2]. Control stations have been developed at DRES and DRDC Suffield since 1982, however have mainly been in use only in unmanned aerial vehicles (UAVs). Slowly around 1990, DRDC Suffield began to implement control stations for use with ground vehicles. Since 1990 then the VCS used by DRDC has been a joint effort between CDL Systems and DRDC Suffield but CDL Systems has also created stations for other customers. The VCS was designed to be independent of the control system and the communications protocol on the vehicle being controlled.

The VCS generally consists of three seperate modules:

1. *Background Map/Mission Planning Module:* This module not vehicle specific. Instead it is a generic module that is used to do the mapping and mission planning of any vehicle.

2. *User Interface:* The user interface is usually vehicle specific. It accesses the mapping module and displays information that is specific to that vehicle.

3. *Data Link Interpreter(DLI)/Vehicle Specific Module:* The DLI is the sensor interpretation module in the VCS. The DLI gathers all of the information from the vehicle being controlled using whatever protocol the vehicle uses and converts that information into a format that the VCS can understand.

In the past the VCS system has been used to support secondary payloads such as an Automated Robotic Scanner (ARS) arm, a backhoe attachment, and a brush cutter. The VCS has been used in the display of geo-registered data from mine detection intrsuments [5]. The VCS has also been adapted to run the ANCÆUS protocol as was required by the improved landmine detection project (ILDP), providing the command and control for the ILDP remote detection vehicle (RDV) as well as real-time

processing software for the four sensory systems. The four sensory systems include an infrared camera (IRC), a metal mine detector (MMD), a ground probing radar (GPR), amd a thermal neutron activation sensor (TNA) [3]. Finally, the VCS has been applied to the CCMAT project and that VCS software comprised of several different applications which interact with each other to control a vehicle. These applications include a scrolling map, a task editor, a data link interpreter, a warning panel, a control panel, the metal detector data processor, the hand held controller and finally the video display [4].

The latest enhancements to the VCS, including the hardware system, simplified the setup and operation of the VCS to enable its use by non-technical personnel with minimil training [5].

In summary, VCS has been designed to primarily be a teleoperated system and for use with semi-autonomous AUVs, but has the capability to control generic unmanned vehicle systems.

2. Technology Audit

2.1 ANCÆUS/AVATAR

The contents of this audit are based upon documentation and source code dated from 1999 to 2001 [7]. The software is composed of N baseline modules and a number of application specific modules written in ANSI standard C and, in later implementations, C++. The software has been designed, with a few exceptions, to be operating system independent. In general the software appears to be well organized, but lacks industry standard commenting. In particular, the code does not list authorship, creation dates, modification authorship and dates of source, header, or make files or functions.

2.1.1 Software Description

The main software elements of ANCÆUS are a *network topology*, a *routing protocol*, a set of generic *control codes*, *vehicle control processor* software, *vehicle personality module* software, and *control station* software.

ANCÆUS is founded upon a simple network topology that exploits *units* and *subunits* analogous to modern network *nodes* and *subnodes*. Data (such as commands to the vehicle or responses from a vehicle) are placed into *packets* of information that can be sent to specific *devices* with a unique `unit.subunit` address.

In implementation, a unit is a collection of subunits on a single vehicle, typically devices on that vehicle. Arguably the most important subunit of any unit, however, is the *router* that passes data to and from subunits. A router is composed of three basic parts

Device scanner This process queries all ’registered’ devices for input packets and can operate in either interrupt driven or polled modes.

System tables Specifically a device table and a routing table. The device table is an essentially static list of devices scanned by the *device scanner*. The routing table is a dynamic list of devices and subunit addresses.

Router handler The router handler, with a subunit address of 0, adds (or ’registers’) to the routing table, removes, perform compression/decompression and interfaces to subunits for diagnostics.

The packet structure permits subunits to communicate to one another over the network through a simple *routing protocol*. This protocol literally determines the route through which a given packet will travel and which subunit is authorized to open it. The packet structure allows the network to address up to 255 units and a further 255 subunits per unit. The packet has the structure outlined in Table 1:

Type	Name	Purpose
uchar	flag	command/response bit
uchar	desthi	8 bit upper 'unit' destination address
uchar	destlo	8 bit lower 'subunit' destinationaddress
uchar	srchi	8 bit upper 'unit' source address
uchar	srclo	8 bit lower 'subunit' source address
uchar	count	number of bytes in data
uchar	data	a data block of up to 128 bytes

Table 1: *ANCÆUS packet structure. The Flag byte has been recently modified to enable up to six communication 'channels'.*

The basic routing algorithm is complicated by the use of a shared frequency among multiple vehicles. Originally, ANCÆUS used a home grown synchronous protocol over a 'dumb' RF modem. The control station would transmit a sequence of packets, one for each vehicle unit. All the vehicles would listen to this sequence, noting the order of transmission and opening only the packet addressed to the vehicle's unit. Responses to the control station would be returned in an identical order, meaning that each vehicle would listen for preceding vehicle transmissions transmitting response packets. Contingencies were established within the protocol to cope with dropped packets, loss of signal, etc. Using a similar proprietary packet collision detection method, the Freewave modems used in later ANCÆUS projects removed the requirement for this home grown protocol.

At the core of the communications protocol is an exhaustive list of generic control codes. The control codes are composed of two parts: a two byte identifier and an argument field. The identifier is composed of a group byte, loosely related to the type of command, and the command code itself. The list is extensive and covers the command types: Network Control, Engine Control, Chassis Control, Electrical/Ancilliary Control, Joystick Control, Navigation Control, Video/Audio Control, Autonomous Control, and Joint Movement Control (for manipulator arms). Some typical examples of command codes include

Network Shutdown. This command that ensures the orderly shutdown of all units and subunits

Engine Start. Platform dependent (e.g. For an Argo: set throttle to 10%, engage starter, wait for 600rpm and disengage starter).

Enquire Vehicle Position This command interrogates a vehicle for its current cartesian position $< x, y, z >$.

There are more than 150 ANCÆUS control codes, though these must be divided into command and response codes. The total number of

observable/controllable values is actually closer to 100.

Given the size of the control codes and packet structure, ANCÆUS will routinely cluster multiple control codes for a single device into a single packet.

The ANCÆUS Vehicle Control Processor houses a network gateway between the control data link and the vehicles internal network.

The ANCÆUS Vehicle Personality Module implements the vehicle specific functions to control and monitor the state of the vehicle.

The ANCÆUS Control Station, originally written for AmigaDOS and later ported to OS/2 v3.0 , now being ported to Windows XP though there is no formal documentation describing this porting effort. The OS/2 control station is composed of a router and three IPC devices: a user interface module, control console/joystick, and a GPS master module. IPC devices are an OS/2 interpocess communications facility that permits a central process (the router) to call upon functions within other processes, similar in many respects to the UNIX RPC and IPC protocols.

2.1.2 Hardware Description

In agreement with the original ANCÆUS concept, current ANCÆUS/AVATAR hardware is composed of a control station, a VPC, a VPM, a navigation module, a MUX/Xmtr Pan/Tilt module and a number of vehicle platforms that have specific software tailored to their capabilities.

The control station is a general purpose computer equipped with keyboard, monitor, and mouse/trackball. Additional elements include a data link, audio/video receiver/mux, a control console, and specialized video display cards.

The Audio/Receiver Mux module receives audio and video from a vehicle and converts it to RS-170. This feed is connected to the computer display. As of 2001, the receiver was an Emheiser 2.4Ghz line-of-sight FM receiver with an approximate 10km range. The current system employs a Dynapix video reciever.

Two data link modules are available to communicate with the vehicles, a Data Radio TM96B modem (2W, FM, fixed frequency, 9600 baud, 4-10km range) and a lower power Freewave modem (300mW, frequency hopping, 115 kbaud, 2-5km range).

The data link modules connect the control station to the tele-operated vehicle platform. The VPC interfaces with the data link module and receives the tele-operation commands from the remote control station. It implements the ANCÆUS router software that allows it to route the received commands to

the appropriate destination. At present, the VPC is implemented on a VME backplane using a Motorola 68030 processors. The current software does not utilize an operating system.

All other ANCÆUS modules are implemented on the Motorola 8 Bit 68HC11 microprocessor. These modules include:

- The Vehicle Personality Module
- The Navigation Module
- The MUX/Xmtr Pan/Tilt Module

All modules on the vehicle are linked via a custom fibre optic loop. The VPC router uses the fibre optic loop and the ANCÆUS protocol to route commands to their appropriate destination. The software for the 68HC11 microprocessors comes in two flavors, with the newer software being written under C while the older software was handcrafted in assembly language.

The preceding paragraphs describe the current ANCÆUS/AVATAR implementation. Via a contract with Amtech the ANCÆUS/AVATAR hardware/software is experiencing a significant upgrade that will extend its capabilities. The Motorola MPC555 microcontroller is being introduced that will replace the existing VPC and VP module by combining these formerly independent hardware pieces into a single microcontroller. The MPC555 microcontroller has been implemented on a custom designed board known as the Generic Robotics Controller. This board includes many features that are useful for robotics control, including the following:

- 2MB RAM, 8MB Flash
- 4 RS232 and 2 CAN Ports
- Fibre Optic data link
- 8 (5 Amp) PWM drivers
- 24 (0.5 Amp) relay drivers
- 4 12 Bit D/A converters
- 15 chassis A/D, 8 PWM A/D feedback channels
- 2 Wheel encoder channels
- 4 variable gain 12 Bit A/D channels

The MPC555 is scheduled to run the CMX real-time operating system and Amtech has undertaken to port the VPC software and the Argo VPM software to run under the CMX real-time operating system. This contract is scheduled to be complete in the near future.

2.1.3 Documentation

Though the state of documentation is much improved since the 1996 ARC commercialization report. Documentation for ANCÆUS/AVATAR remains incomplete and dated. Given the complexity of ANCÆUS, the volume of work required to fully document the system has overwhelmed the project staff.

The most current documentation [7] is a reference document describing the elements of ANCÆUS as of 2001. The document does not directly treat ANCÆUS/AVATAR and only alludes to a conversion to Windows. The document provides some insight into the network topology, router operations, and control codes. The document does not describe implementations of the 'personality' modules. This module, or VPM, represents a significant amount of work for a given vehicle conversion, yet there is no reference to methodology or functional specification of a VPM. Certainly, the command code group identifier mentioned earlier implies that a given set of control codes are implemented on a unique device and the documentation does define the desired function of a given control code, but this constitutes a derived functional specification. Whether this lack of specification is intentional is very unclear.

The source code for the VPM for each vehicle controlled is available. Some VP modules are written in C, while others are implmented in assembly language. The Argo, Cat, Hydrostatic, and ILDP vehicles are in C and seem to represent the most recent implementatinos.

2.1.4 Intellectual Property

The IP that underlies the ANCÆUS/AVATAR software is owned by DRDC. However, recent developments to standardize the VPM around a custom designed MPC555 board have granted some hardware IP rights to Amtech.

2.2 Vehicle Control Station

2.2.1 Hardware

Currently the VCS has been adapted into several different hardware platforms and sizes. These include:

1. Lunch Box
2. Humvee
3. Backpack
4. Stationary Station

All VCS systems include some sort of hardware package, regardless of size. Normally no keyboard is used, instead there are programmable push buttons, programmable toggle switches, trackballs (mouse) and joysticks. All the hardware is outlined very professionally in a specific hardware configuration document.

2.2.2 Software

The VCS system is essentially a large software package. Although, since interaction with the system is necessary, hardware controls as outlined in 2.2.1 are used and interact with the software system.

The VCS system is written in C/C++ and Qt and currently operates on the following operating systems: RedHat Linux 7.2 and 7.3, SUN Solaris 2.6, 2.7, (7), 2.8 and HP-Unix 10.10, 10.20 [2]. Qt is a GUI toolkit / class library that can be used with Unix, X11, and Windows systems. Toolkits such as QT are extremely important for developing GUI bases applications on Unix systems because traditionally most Unix windowing systems, like X window, lack the high level programming tools that are traditionally found on most Microsoft Windows and Mac OS systems. On Unix systems graphics capabilities are much more primitive. There are no tools for building GUI apps available. Therefore toolkits such as QT, Motif, etc. are invaluable. Qt is relatively fast, easy to use and portable. QT has already found it's way into various Unix/Linux applications [11].

The VCS is a multi-process system, however does not run as a real time operating system (RTOS). This means that the VCS will not provide any guaranteed latency. However, it is noted that since the VCS is largely a system that requries human interaction, i.e. noticing a red-light or the pressing of a stop button, latency isn't necessarily an issue since any human reaction to an event will be considerably slower than the current system would react. i.e. a 10-50ms delay can be expected but no specific time is guaranteed.

Note: According to [2] some of the key features of the VCS are:

- Real time situation awareness
- Real time vehicle and payload control
- Real time data fusion and GIS coupling

The VCS, as referenced in 2.2, can be and has been configured for several applications. The VCS has been adapted to work with all of the current ANCÆUS systems for vehicle control etc.

The VCS currently does not have "real" multiple robot ability. Instead the VCS must have multiple sensor interpretors or DLI's in order to talk to each

of the robots being controlled. Also the user would only be able to control or monitor a single robot at a time. However, the VCS has been adopted to control more than one unmanned vehicle but the DLI hardare and software was preprogrammed to know how many vehicles were going to be controlled. i.e. target systems handles 3 UVs.

2.2.3 Documentation

With each new release it appears that CDL Systems has provided very comperehensive documentation regarding the VCS. Manuals such as setup procedure, software design document, hardware configuration, software configuration, product release document, and a final report are all available for the CCMAT project. Not only are these documents available that all are written very clearly and concisely. All documents are laid out in a very professional manner using pictures where necessary etc.

2.2.4 Business

DRDC Suffield and CDL Systems each own different modules of the VCS. VCS is licensed by DRDC Suffield to CDL Systems. However, CDL Systems has written additional modules and interfaces of which they are the sole owner. Thus, DRDC Suffield is allowed to view these modules and build on them, but not allowed to license them to another company. For example, if DRDC Suffield wanted a university to do some work on the VCS, instead of giving them source code for the VCS, DRDC Suffield would only be able to provide an API.

DRDC Suffield in the past 5 to 10 years has relied heavily on CDL Systems ability to create the VCS that is now in use at DRDC Suffield. Thus, to move forward with the current VCS, either CDL Systems will have to be given a contract to complete the changes and improvements necessary or several defense scientists will have to commit their time to learning about the current VCS and reading through the years and years of source code for the VCS.

CDL Systems has 28 employees who have put in over 100 man years into the VCS system. Unfortunately that means we have a large dependance on CDL Systems to keep the VCS in a state that benefits our needs. However, since CDL Systems is largely in charge of the VCS development they are also the point of contact for any possible technical support or training issues.

2.3 Alternative Technologies

There are many operating systems and software applications that could be used to create autonomous mobile robotic platforms in addition to the teleoperation applications. A survey of these found several candidate systems that should be

considered as alternatives to the ANCÆUS system. The selection of alternatives was limited to full-blown and operational autonomous mobile robotic architectures rather than teleoperation. The background to each of these candidates considers the following questions:

1. Is it an operating system?
2. What software model does it use if any?
3. What communication protocol does it use if any between components?
4. What robot control architecture does it use/specify if any?
5. Is it proprietary or open?
6. Where has it been applied?
7. Was it meant for a single robot control or multiple robots?
8. Can it be used easily with simulators?

2.3.1 Saphira/OAA

Saphira was the operating system created at Stanford's SRI for the Erratic and Pioneer robots [13]. It was coupled to SRI's Open Agent Architecture to tackle multi-robot systems and in 1996 won the AAAI *Hold a Meeting* autonomous robot competition [12] . This system was used on Real World Interface B series robotics and the ActivMedia Robotics Pioneer series which are the commercial versions of the original Stanford robots. Saphira is a proprietary operating system that is free (older version) for educational purposes. The newer object oriented version, Aria, is proprietary software sold with ActivMedia robotics. At the core of the system is the Local Perceptual Space (LPS), which stores world representation as geometry. The robot control architecture that is used is a hybrid of the reactive fuzzy logic-based behaviours and the higher level PRS-Lite deliberative reasoning system. Saphira uses the Client/Server model and has many devices that can be attached to the robots with drivers. Spahira was intended for one to one control and was extended to multi-agent systems by using Open Agent Architecture. Open Agent Architecture can use object-oriented or monlithic modules alike in a distributed agent environment. OOA uses the Interagent Communication Language as the communication protocol. The Open Agent Architecture is free to non-commercial purposes under the OAA Community License. The Saphira systems has an internal simulation interface so that experiments can be conducted on real platforms or on simulated ones.

2.3.2 Player/Stage

Player/Stage started out as Golem/Arena at USC, and has developed into an open source project for the simulation and operation of various mobile robot platforms. Player is a multithreaded robot device server and Stage is the simulation environment. However, the simulation module can itself be removed and the robot server can be run on a real robot without changing any software. This is done by changing the configuration file associated with the intended robot. Player is meant as a platform and language independent system, although it is primarily written in C and it is intended for the GNU gcc compiler. This system allows the user to use separate robot device interface/driver so that the client software sees all robots as identical modules and leaves the interpretation of the commands to the specific platforms. Player/Stage is object-oriented, using the device context from UNIX to signify abstract robot entities. Player uses the Client/Server model but allows multiple access by multiple clients simultaneously so is not limited to direct one to one relationships. Player/Stage uses TCP as its communication protocol so it is inherently networked without further effort. Player/Stage does not define a robot control architecture for operation. This software is free for all uses under the GNU Public License (GPL) and is funded by DARPA(MARS). The open source community for this software includes many universities and research organizations from across the United States and Canada.

2.3.3 CLARAty

CLARAty (Coupled Layered Architecture for Robotic Autonomy) is the realisation by NASA/JPL that they have many programs using diverse operating systems and robot control architectures in parallel and in conflict without standardization. Richard Volpe and others [15, 8, 10, 9] proposed an object-oriented approach to robotic control and have successfully adapted this operating system to currently-fielded systems like Rocky 8, FIDO, K9, Rocky 7 and the ATRV rovers. It is a complete operating system that combines the traditional planning and executive layers as a common layer above the functional operation of devices and platforms. It uses abstracted objects signifying individual subcomponents as well as entire robots that work together from a common command set. The CLARAty system allows the insertion of simulated components or emulators which do not change the interface between the simulated components and the system. This is more advantageous than simply running a simulator as the replacement can occur at below the robot level. JPL and NASA want to open the CLARAty architecture as an open source project but are running into resistance from the US Department of Commerce. However, in consultation with Issa Nesnas, as Canadian defence researchers we would be able to obtain licenses and open source if we decided to adopt this option. The exact method to achieve this is yet to be determined.

2.3.4 CARMEN

CARMEN (Carnegie Mellon Robot Navigation Toolkit) is a navigation toolkit developed by researchers at The Robotics Institute at Carnegie-Mellon University. It is not a complete operating system, but is a tookit of commonly used robotic routines that operate together as a package to control a robotic platform. It is a open source project and can currently control the iRobot ATRV, iRobot ATRVjr, iRobot B21R, ActivMedia Pioneer I, ActivMedia Pioneer II, Nomadic Technologies Scout, and Nomadic Technologies XR4000. CARMEN has been run on the Red Hat 5.2, 6.2, 7.1, 7.3, 8.0 and SuSE 8.0 versions of Linux.

Carmen provides an interface to a SICK scanner, the capability to build a world representation, a navigator to allow the robot to traverse its world, and the capability to send commands to control the robotic platform on which it is situated.

A key component of CARMEN is its use of InterProcess Communications (IPC) library. IPC is a platform independent package for distributed network-based message passing. It provides facilities for both publish/subscribe and client/server applications and can efficiently pass complicated data structures between different computers.

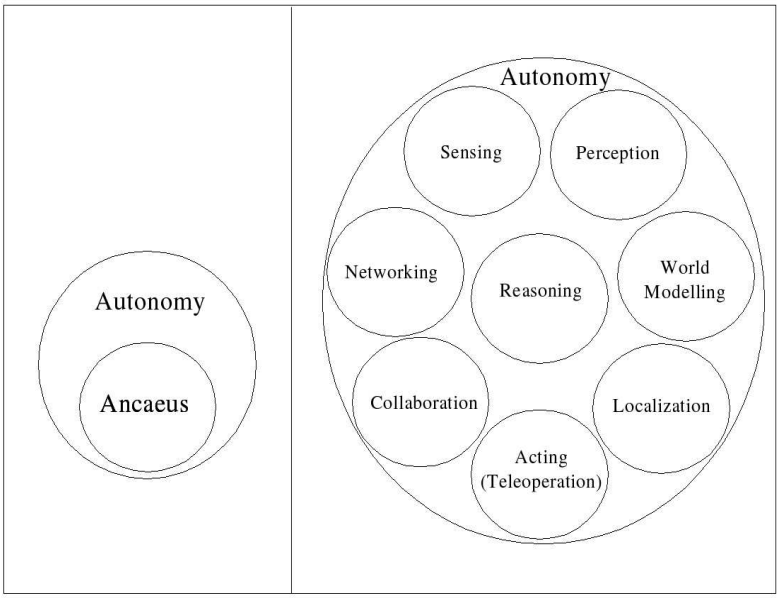


Figure 1: *The Autonomy Myth - that an autonomous system is a simple extension of a teleoperated system.*

3. Discussion

The question of whether ANCÆUS is an enabling technology for autonomy assumes that teleoperation is a central or core capability requirement prior to acheiving autonomy (see Figure 1). This is a contreversial assumption, particularly since the volume of effort and technology required to achieve autonomy is substantially larger than teleoperation. Indeed, many of these elements may require substantial bandwidth within and between vehicles and control station and generally require higher resolution sensing.

3.1 Design Requirements

The ALS program seeks to create a flexible, extensible, and maintainable software architecture that spans a variety of vehicle configurations and applications from skid/steer off-the-shelf vehicles to custom legged platforms and from pure teleoperation to complete autonomy.

ALS is not alone in this effort. A number of defence, industrial, and academic institutions are pursuing similar lines of investigation and face similar requirements, summarized more formally as:

flexibility the ability to apply the architecture to a variety of mechanical configurations and application environments.

extensability the ability to extend the architecture's capabilities without changing software or hardware.

maintainability the ability to easily maintain and debug the baseline system.

modularity the ability to add/remove capabilities without affecting other systems.

portability the ability to move the software from one hardware environment to another.

robustness the ability of the system to withstand adverse conditions.

These broad requirements are further limited by additional requirements that include real time performance, networking capability, off-the-shelf hardware, and the use, where possible, of nonproprietary hardware and software platforms. For our future mult-robot autonomous robotic system, a set of criteria were determined to be necessary for the software, hardware, networking and business aspects.

3.1.1 Software Criteria

- Real-time OS – vehicle platform control rates must be guaranteed.
- Modular – the entire software architecture must be formed of easily interchangeable elements.
- Scalable – code should port from low to high level processing devices with simple changes.
- Open Source – for validation and verification, the entire code base should be open to inspection.
- Portability – code must transfer between various computers/microcontrollers with little or no changes.
- Environment – the programming environment supports various paradigms (e.g. OO methods).
- Device drivers – the hardware should be abstracted, hiding details of specific hardware from higher levels of the application.

3.1.2 Hardware Criteria

- Networked – common networking capabilities (TCP/IP or Ethernet) will ease adoption of new sensors and actuators.
- Integrated video and data – the system software must provide integrated audio, video, command, and data in the same communications channel.

3.1.3 Networking Criteria

- Asynchronous – asynchronous transmission increases the yield of limited bandwidth.
- Self-discovery – dynamic host configuration simplifies robot software design.
- Information Yield – the network protocol must maximize communication bandwidth of each robot.

3.1.4 Business Criteria

- Open source – For comparative research, open source offers reproduceability, code-sharing, debugging, and versioning discipline.
- IP Encumberances – Premature adherance to proprietary solutions can badly skew the research to support the specific technology and/or commercial interests.
- Compliance – leverage other programs through the adoption of existing software standards, such as Player/Stage/Gazebo, Carmen, and Linux.
- COTS hardware – commercial off-the-shelf hardware and software minimizes effort required to add new components to a system.

With these factors in mind, the group has evaluated ANCÆUS and VCS. The following sections will discuss the methods of evaluation, advantages and disadvantages of these systems.

3.2 ANCÆUS/AVATAR

A number of ANCÆUS features reveal the forward thinking of DRDC in robotics development. Some examples include:

Network architecture Network architectures facilitate distributed computing, greatly reducing the cost of computing resources through the use of less complex resources while improving the system's hardware robustness.

Vehicle Modularity Isolation of vehicle specific hardware reduces the system changes between vehicles, isolates vehicle specific characteristics, and eases maintainability and portability of core software modules.

Hardware self discovery Hardware Self Discovery greatly simplifies vehicle configuration and frees users to reconfigure vehicles as necessary without recompiling or recoding system software and without the maintenance and versioning of configuration files.

Multi-robot Autonomous Robotic System Components (Terminology)	
Sub-Component	***Term Definitions***
Control Station (CS)	The Computer system that houses the multimodal interface to the human operator.
Communications Link	The wireless inter-robot communication system for the transmission of commands, data, audio, and video between robots and the operator.
On-Board Computer (OBC)	The higher-level computer responsible for deliberative planning and perception that maintains the inter-robot communications.
Vehicle System Controller (VSC)	Computer/microcontroller unit that is in charge of the real-time control of the robotic vehicle.
Sensors/Transducers	Sub-components whose function is to gather data about the world around the robot.
Actuators	Sub-components whose function is to control adjust the state of the sensor and/or individual elements of the robotic vehicle.
MicroController Unit (MCU)	Microcontrollers responsible for control of modular sub-components on the robotic platform.
Function-Specific Processor (FSP)	Computer/microcontroller elements that are responsible for carrying out medium/lower level computations/processing and communicating

Table 2: *Hardware terminology for a basic autonomous architecture.*

Multivehicle Networking Multivehicle protocols are essential for cooperative robotics, particularly in a military environment. This protocol must cope with communication dropouts and provide message relaying like any modern network.

ANCÆUS documents reveal that these were the philosophical design objectives embodied in the system. Though conceived over 15 years ago, each of these ideals are equally desireable today. Indeed the anticipated structure of the ALS platforms (see Table 2) closely mirrors the ANCÆUS concept. Unfortunately, these same leading edge features comprise the central problem of ANCÆUS as an enabling technology for future work in autonomous systems.

The ANCÆUS network is founded upon a simple, but unique network packet. Not surprisingly, the protocol and hardware that handles these packets is also unique to ANCÆUS . When ANCÆUS was created in the early 1990's the TCP/IP protocol was relatively new and would have been very slow, inefficient, and expensive to implement as a vehicle network. The development of a custom lightweight networking system was a logical choice for the ANCÆUS team. Further, video and audio traffic could only be supported by independent RF channels since wireless networking capable of audio and video streaming did not exist.

However, over the intervening decade, TCP/IP hardware and software have become more powerful, widely available, very robust and inexpensive. Further, compatible

industrial protocols for low-level hardware have emerged, such as CANBUS and DEVICENET. Given the cost, performance, and versatility of these systems, it is difficult to justify the use of a limited and more expensive proprietary vehicle networking system. With the arrival of broadband wireless in the late 1990's the ability to simultaneously stream data, audio, and video has become possible using off the shelf hardware.

Replacing the networking elements of ANCÆUS with modern protocols substantially reduces the applicable scope of ANCÆUS technology but would not diminish the required software support. For example the control station software, vehicle command processor logic, vehicle personality module logic would all require conversion to remote procedure calls, a more efficient interprocess communication mechanism, but also a substantial body of work.

The effect of technological change can be seen in the adoption of the Freewave modem. The Freewave modem provided multivehicle communication 'out of the box' and removed a substantial block of multivehicle logic custom written for ANCÆUS using 'dumb modems'. The adoption of TCP/IP would have a similar, though more profound impact on ANCÆUS. Most of the generic ANCÆUS components would be replaced by industry standard TCP/IP hardware and software. The remains of ANCÆUS would be reduced to an API of custom RPC calls to dedicated hardware devices – work that would have to be done to support any networking system including modern ethernet compliant methods such as CMU's IPC.

Specifically, this study reveals a number of technical obstacles to ANCÆUS as an enabling platform:

1. Separate Communications Links- separate audio/video and command and control links between control station and robots. Consolidation into single link would require major reengineering of ANCÆUS to use TCP/IP.
2. Networking - uses a proprietary communications protocol currently incompatible with other COTS networking equipment using TCP/IP, though the ANCÆUS packet could be transported by TCP/IP. However, doing so would render the ANCÆUS device addressing feature redundant and reduce ANCÆUS to an 'application server'. Significantly, proven robot application servers are available through open source channels.
3. Fibre Optic Link - Intravehicle communications is via nonstandard fibre optic token ring. Since the optical network is a ring (as opposed to star) topology, the failure of a single device can interrupt the communications to all devices on the network.
4. Teleoperation - Human finesse in vehicle control i.e. lower precision vehicle requirements.
5. Quality - while some elements of the system are well concieved and crafted, user, reference, and source code documentation are out of date and well below industry standard. This will greatly hamper adoption and use of ANCÆUS by third parties.

These points describe the current technical obstacles that exist within the ANCÆUS operating system specifically that would require significant effort to remove. Most of these obstacles represent critical shortfalls and would need to be addressed prior to the 2005 demo.

3.3 VCS

As mentioned in 2.2.4 DRDC Suffield has relied heavily on CDL Systems creation of the VCS, to the point that over 100 man years have been invested in this program by CDL Systems. The dependence on VCS and proprietary encumberances undermine the suitability of VCS for future multi-robotics projects.

Quoting [1], VCS brings the needed capabilities of unmanned vehicle and surveillance systems into a fully integrated, configurable and cost effective package. However, VCS has not been adapted for use in a multi-robotics system, ensuring that significant effort is necessary to adapt VCS to Autonomous Land Systems (ALS) ambitions. Further licensing 2.2.4 requires either CDL Systems or defense scientists at DRDC Suffield to make these changes. Additional agreements would be necessary for CDL Systems to make necessary changes. Conversely, DRDC Suffield would experience a very high learning curve prior to making significant changes to the system.

3.4 Comparitive Review of Alternative Technologies

In order to compare the relative merit of the candidate systems under inspection, a list of MRAS capabilities was developed. The tables define 32 separate areas of multi-robot autonomous robotic system capability that are required to fulfill the highest levels of autonomy. The precise delineation of the various fields is admittedly arbitrary, with some sub-fields being worthy of equal merit. However, for the sake of the argument, these fields represent a reasonable dissection of the entire autonomy problem.

Table 3 estimates the current state of the operating systems in question as measured by these 32 capabilities. The value is ranked from 0, meaning no capability in the specific field, to 10, a complete capability in that specific field and requiring no future expansion or redesign. The total ranking of a complete, ideal system would be 320 points. Far from absolute or precise, this ranking is meant as an approximate, relative assessment of capabilities. The totals for each operating system appear at the bottom with their relative percentage in comparison to an ideal system. It can be seen that no system under inspection has sufficient capabilities to require no further work. It can also be seen that, in general, the teleoperation systems ANCÆUS and VCS lack capability in relation to wider scoped commercial/open source autonomous operating systems.

In a similar manner, Table 4 describes the *effort* required to add or improve capability to a level sufficient to conduct MRAS research. The ranking ranges from 0 which would require minimal or no effort to improve the operating system to 10 which would require complete design/redesign of capabilities. The worst-case effort required would

be a complete design from scratch of an autonomous robotic operating system and this would score a total of 320 in the respective capabilities. In general, it is clear that a majority of the work is needed to reach multi-robot autonomous robotic system research from any alternative at this point in time.

Table 5 describes the autonomous capability areas that each level of autonomy require as a minimum. The autonomous capability for teleoperation requires the minimum of areas to construct a complete system. Full autonomy requires all 32 autonomous capabilities. As expected, the number of autonomous capabilities needed increases for the intervening two levels of autonomy: safe mode and shared control.

Table 6 estimates the current state of each alternative technology and the effort needed at each level of autonomy. Regardless of software technology, the level of effort increases with the desired level of autonomy. It is not surprising, therefore, that the commercial/open source systems are generally more capable than ANCÆUS, the result of a relatively small development effort focussed on teleoperation.

Table 7 describes qualitative scientific and programmatic factors that are advantageous to the ALS program. The software technologies under inspection here denote whether they have these advantages or not. The two open source operating systems, Player/Stage and CARMEN, are the only alternatives that embody all advantages.

In another view, Tables 8 and 9 below complete templates 1 and 2 posed in the Engineering Review Terms of Reference.

Multi-robot Autonomous Robotic System Capabilities (Current State)					
Capability	***Ancaeus***	***Player/Stage***	***Carmen***	***Saphira/OAA***	***VCS***
	Current	Current	Current	Current	Current
GIS/ Map Data Conversion	2	0	0	0	3
Vehicle Dynamic Modelling	0	4	5	5	2
Real-time Control (Actuators)	10	8	8	7	10
Teleoperation	10	8	6.7	6	10
Control Architectures	3	8	4.7	4	1
Robotic Locomotion	10	8	5.3	7	2.7
Sensing (Transducers)	10	6	8	8	6.7
Perception (Machine Vision, Sensor Fusion etc.)	0	5	4	4	2.7
Localization (Nav Filter, SLAM etc.)	2	5	7	4	0.7
World Modelling (Mapping etc.)	1	5	4.7	5	2.5
Navigation/ Motion Planning	3	8	8	8	2.5
Collision Avoidance/ Recovery	0.6	6	5.3	3	1
Intra-Robot Communications	5.5	8	8	8	2
Inter-Robot Communications	5	8	8	8	3.75
Networking Connectivity/ Quality of Service Protocols (Low Bandwidth/ Over Horizon)	1	2	2	2.5	3
Collaborative Human/Robot Perception	0	2	2	2	0
Collaborative Human/Robot Planning and Resource Allocation	0	2.7	2.7	2.7	2
Shared Situational Awareness (Operator/Robot)	4	3	4	4	2
Behaviours/Skills	2	7	7	7	3
Cognizant Failure/Learning/Adaptation	2	2	2	2	1.3
Multi-modal Interface	3	3	8.7	8	5.3
Health Maintenance	7	4	2.3	5	1.7
Sliding Autonomy	0	2	2	2	2.7
Human-Robot Cooperation	1	1	1	1	1
Learned Trafficability	0	0	0	0	0
End-Effector Manipulation	3.75	0.5	0	0	3.3
Symbolic Knowledge	0	1.3	1.3	1.3	1.3
Uncertainty and Reasoning	0	0	0	0	0
Problem Solving	0	0	0	0	0
Multi-modal Control Station	4	5	3.3	6	8.3
Natural Language Processing	0	0	0	0	0
2D Simulation	1	10	8.3	6	1
3D Simulation	0	7	0	0	1
Totals	***90.85***	***139.5***	***129.3***	***126.5***	***87.45***
Percentage of 320 %	***0.28***	***0.44***	***0.4***	***0.4***	***0.27***

Table 3: *A subjective assessment of the current state of ANCÆUS, VCS, and some commercial/open source systems in the MRAS context.*

Multi-robot Autonomous Robotic System Capabilities (Effort Needed)					
Capability	***Ancaeus***	***Player/Stage***	***Carmen***	***Saphira/OAA***	***VCS***
	Effort	Effort	Effort	Effort	Effort
GIS/ Map Data Conversion	10	10	10	10	10
Vehicle Dynamic Modelling	10	2	6.7	6.7	8
Real-time Control (Actuators)	4	4	6	4	4
Teleoperation	4	5	6	8	4
Control Architectures	7	2	5.7	7	10
Robotic Locomotion	5	5	6	3	8.3
Sensing (Transducers)	4	4	4	4	4
Perception (Machine Vision, Sensor Fusion etc.)	10	10	8	8	8
Localization (Nav Filter, SLAM etc.)	10	10	5	6	7.6
World Modelling (Mapping etc.)	10	5	5	6	10
Navigation/ Motion Planning	10	5	5	5	10
Collision Avoidance/ Recovery	8	5	4	3	9
Intra-Robot Communications	10	2	2	2	10
Inter-Robot Communications	10	2	2	2	10
Networking Connectivity/ Quality of Service Protocols (Low Bandwidth/ Over Horizon)	10	5	5	5	10
Collaborative Human/Robot Perception	10	8	8	8	10
Collaborative Human/Robot Planning and Resource Allocation	10	8	8	8	9.3
Shared Situational Awareness (Operator/Robot)	10	8	8	8	9.3
Behaviours/Skills	8	6.7	8	10	8.5
Cognizant Failure/Learning/Adaptation	8.3	10	10	10	8.8
Multi-modal Interface	7	5	3	6	10
Health Maintenance	3	6	7.3	9.7	10
Sliding Autonomy	10	10	10	10	10
Human-Robot Cooperation	10	10	10	10	10
Learned Trafficability	10	10	10	10	10
End-Effector Manipulation	6.5	10	10	10	10
Symbolic Knowledge	10	10	10	10	10
Uncertainty and Reasoning	10	10	10	10	10
Problem Solving	10	10	10	10	10
Multi-modal Control Station	10	7	8	10	10
Natural Language Processing	10	10	10	10	10
2D Simulation	10	3	10	5	10
3D Simulation	10	5	10	10	10
Totals	***242.8***	***132.7***	***180.7***	***155.4***	***278.8***
Percentage of 320 %	***0.76***	***0.41***	***0.56***	***0.49***	***0.87***

Table 4: *A subjective assessment of the required effort to provide ANCÆUS, VCS, and some commercial/open source systems with MRAS capabilities.*

Multi-robot Autonomous Robotic System Capabilities (Capabilities vs. Autonomy Level)				
Task	***Teleoperation***	***Safe Mode***	***Shared Control***	***Full Autonomy***
GIS/ Map Data Conversion				X
Vehicle Dynamic Modelling		X	X	X
Real-time Control (Actuators)	X	X	X	X
Teleoperation	X	X	X	X
Control Architectures	X	X	X	X
Robotic Locomotion	X	X	X	X
Sensing (Transducers)	X	X	X	X
Perception (Machine Vision, Sensor Fusion etc.)		X	X	X
Localization (Nav Filter, SLAM etc.)			X	X
World Modelling (Mapping etc.)			X	X
Navigation/ Motion Planning			X	X
Collision Avoidance/ Recovery		X	X	X
Intra-Robot Communications	X	X	X	X
Inter-Robot Communications	X	X	X	X
Networking Connectivity/ Quality of Service Protocols (Low Bandwidth/ Over Horizon)			X	X
Collaborative Human/Robot Perception			X	X
Collaborative Human/Robot Planning and Resource Allocation			X	X
Shared Situational Awareness (Operator/Robot)			X	X
Behaviours/Skills		X	X	X
Cognizant Failure/Learning/Adaptation				X
Multi-modal Interface				X
Health Maintenance	X	X	X	X
Sliding Autonomy	X	X	X	X
Human-Robot Cooperation			X	X
Learned Trafficability				X
End-Effector Manipulation	X	X	X	X
Symbolic Knowledge				X
Uncertainty and Reasoning				X
Problem Solving				X
Multi-Modal Control Station	X	X	X	X
Natural Language Processing			X	X
2D Simulation	X	X	X	X
3D Simulation	X	X	X	X

Table 5: *Relating the 'levels of autonomy' to the MRAS capabilities list and using the assessment in 3 to provide a subjective measure of autonomy for ANCÆUS, VCS, and commercial/open source systems.*

Multi-robot Autonomous Robotic System Capabilities (Effort vs. Autonomy Level)								
Alternatives	***Level of Autonomy***							
Alternative Operating Systems	Teleoperation		Safe Mode		Shared Control		Full Autonomy	
	Current	Effort	Current	Effort	Current	Effort	Current	Effort
Ancaeus	5.36	7.8	3.11	8.4	2.21	9.2	1.69	9.26
Player/Stage	6.43	6.53	6.22	6.22	4.7	6.94	3.79	8.11
Carmen	4.96	7.31	5.2	6.99	4.34	7.05	3.58	8.03
Saphira/OAA	5.15	7.28	4.91	7.1	4.07	7.22	3.39	8.32
VCS	4.16	9.25	3.2	8.81	2.36	9.19	2.33	9.51
Average	***5.21***	***7.63***	***4.53***	***7.5***	***3.54***	***7.92***	***2.96***	***8.65***

Table 6: *Combined results comparing the current state and effort needed for representative commercial/open source systems within the levels of autonomy.*

Scientific and Programmatic Advantages				
Capability	***Ancaeus***	***Player/Stage***	***Carmen***	***Saphira/OAA***
Verifiable- Ability to verify or compare results through use of common systems used by other researchers	No	Yes	Yes	Yes
Interoperable- Ability to use US research and sub-components within system	No	Yes	Yes	Yes
Expandable- Ability to port current sub-components to other vehicles and applications	Yes	Yes	Yes	Yes
Open Source- Free and unfettered access to source code	No* (Possible)	Yes	Yes	No
Commercial Encumberance- Ability to give system sub-components to 3rd party companies and/or universities to work on contracts with minimal/no cost to us or 3rd party	No (Amtech)	Yes (Open Source)	Yes (Open Source)	No (ActivMedia Robotics)
Inexpensive- Minimal effort needed to add new sub-components to system or use existing ones	No (Effort to make drivers)	Yes (Freely available drivers for new hardware)	Yes	No(Some drivers available under software support)
Synergy- Ability to leverage work by other laboratories using existing software sub-components made elsewhere	No	Yes	Yes	No
COTS Hardware- Ability to plug and play existing/future hardware based on common architectures	No	Yes	Yes	Yes

Table 7: *Scientific and Technical Advantages of ANCÆUS and commercial/open source systems.*

Technical Perspectives		
Advantages		
Ser.	Description	Justification
1	20 years of experience	Evolution of ANCÆUS has allowed for a great deal of refinement in comparison to more recent technologies.
2	Small packet size	Efficient communication for control applications.
3	Hardware modularity	The abstraction of hardware makes all implementations generic
4	Networked architecture	The design allows for mulitple operator control stations and/or multiple robots.
Disadvantages		
1	Separate Communication Links	Separate audio/video and command and control links between control station and robots requiring two sets of modems. Requires multiplexing HW on all robots to share audio/video.
2	Networking	Modifying ANCÆUS for high bandwidth video streaming/relaying and interrobot world model resolution would require substantial development.
3	Fibre Optic Link	Intravehicle communications is via nonstandard fibre optic token ring (brittle to failure).
4	Teleoperation Premise	Human finesse in vehicle control i.e. lower precision vehicle requirements.
5	Limited Packet Size	Small fixed packets preclude audio and video packet and large data transfers like symbolic world models
5	Quality	User, reference, and source code documentation are out of date and below industry standard.
Business Perspectives		
Advantages		
1	Proven Technology	Field proven, robust hardware.
2	Return on Investment	Additional returns from long term investment
Disadvantages		
1	Proprietary	Currently not COTS compatible, meaning effort needed to write drivers for every new device .
2	Significant Effort Needed	Work to evolve ANCÆUS to MRAS far greater than modifying existing autonomous OS.
3	Increased Program Risk	Redesigning for MRAS would involve development and testing risks in contrast to adopting an autonomous system standard.

Table 8: *Template 1 List and comment upon the technical and business advantages and disadvantages of adopting ANCÆUS as the default enabling technology for platform remoting in DRDC ALS program.*

ALS Platform	*Level of Autonomy*								*Comments*
	Teleoperation		Safe Mode		Shared Control		Full Autonomy		
	HW	SW	HW	SW	HW	SW	HW	SW	
ANT	Y	Y	Y	Y	N	N	N	N	
PAW	Y	Y	Y	Y	N	N	N	N	
XTB	Y	Y	Y	Y	N	N	N	N	
HATV	Y	Y	Y	Y	N	N	N	N	
ARGO	Y	Y	Y	Y	N	N	N	N	
BADGER	Y	Y	Y	Y	N	N	N	N	
CAT	Y	Y	Y	Y	N	N	N	N	
HYDROSTATIC	Y	Y	Y	Y	N	N	N	N	
ILDP	Y	Y	Y	Y	N	N	N	N	
ROMMIDS	Y	Y	Y	Y	N	N	N	N	
TRAM	Y	Y	Y	Y	N	N	N	N	
SRV	Y	Y	Y	Y	N	N	N	N	
TTV	Y	Y	Y	Y	N	N	N	N	
Pioneer	N	N	N	N	N	N	N	N	
Segway	N	N	N	N	N	N	N	N	
Helicopter	Y	Y	Y	Y	N	N	N	N	

Table 9: *Template 2: For each ALS platform, does ANCÆUS provide a sufficient hardware and/or software infrastructure to remote the platform to the specified level of Autonomy?*

4. Conclusions

Clearly a forward thinking technology development program, ANCÆUS solved difficult wireless networking and vehicle control problems five years or more before similar systems appeared internationally. Further, ANCÆUS demonstrated the viability of network architectures for fieldable robotic systems (a concept central to many new robot controllers including the ALS future architecture) and demonstrated multivehicle control.

Unfortunately, the networking requirements for the future ALS architecture exceed current ANCÆUS capabilities. Though the ANCÆUS packet is simpler than comparable TCP/IP packets, high bandwidth applications such as video streaming/relaying and interrobot world model resolution would require substantial development of the ANCÆUS networking protocol, largely duplicating existing TCP/IP methods.

Of course, ANCÆUS could be overhauled, adopting TCP as the base transport layer. This would recast the router as an application server and device subunits as clients, a direction ANCÆUS is clearly heading in the merging of VPM and VCP modules. This is a promising direction, so much so that proven TCP application servers have already been developed for robotic systems [14].

Regardless of the application server architecture, ANCÆUS has developed a unique, compact, and proven command set for internal combustion engine vehicles. It is possible that this command set could be used as a starting point for future APIs and extended to include electric, airborne, and water borne vehicles. ANCÆUS now resides on a large variety of remarkable, capable vehicles.

Autonomous multivehicle control also presents some difficulties to both the VCS and ANCÆUS control stations. Both were designed with single vehicle control in mind and both use ’attention swapping’ to achieve multivehicle control. The multivehicle operations proposed by ALS would require a significant reassessment and redesign of both packages.

While VCS has been modified to work with ANCÆUS, the software is proprietary. Additions/changes to VCS to support intended ALS operations would be costly and would further confuse the intellectual property relationship between CDL Systems and DRDC. The ANCÆUS control station, by comparison, has no proprietary encumberances, but is a far less capable control station and even less amenable to large scale multivehicle control.

In terms of hardware, the ALS program foresees the growth of networked ’smart’ sensors adhering to CAN bus, TCP/IP, or IEEE-1394b protocols. The ANCÆUS team has also observed these trends and is experimenting with a CAN bus conversion in conjunction with a new MPC555 microcontroller from Amtech, consolidating VPM and VCP onto one board. These efforts show that ANCÆUS is not a fixed

hardware/software target, greatly complicating the assessment of ANCÆUS as an enabling architecture.

In spite of these uncertainties, a number of options are possible. The following section will outline these options and recommend a course of action.

4.1 Recommendations

From this report a number of options can be identified:

1. Adopt ANCÆUS to control all low-level ALS platforms and then develop/adopt a high-level software technology to work above ANCÆUS for full autonomy.

2. Overhaul the ANCÆUS system to include high-level software technology to be capable of full autonomy.

3. Adopt/develop an alternative software technology using the ANCÆUS control codes for all ALS platforms.

4. Adopt/develop an alternative software technology but maintain a low-level ANCÆUS API for legacy ANCÆUS platforms.

5. Adopt an existing alternative software technology and hardware standards to support a new ALS architecture.

6. Build a new ALS software technology according to a new design philosophy from the ground up.

Adopting option 1 would require no changes to ANCÆUS in its present state. Instead, a new layer would be needed on top to control multiple robotic platforms in a MRAS. This would add an additional communication layer between the higher level software and the low-level ANCÆUS . This option would allow the use of all legacy platforms without modification. The networking overhead for this option would effectively be doubled and would impact the capabilities of the final system to manage information exchange. It would also double the effort to maintain two separate software technologies concurrently. Since inter-robot and intra-robot communication are critical to the success of autonomous systems this is felt to be a critical technical obstacle. This option is not recommended.

Adopting option 2 would require a complete rewrite of the ANCÆUS technology to meet the needs of full autonomy. If this option was undertaken, the advantages of the tried and tested ANCÆUS are nullified and the code base would be open to the same level of risk as developing a completely new software technology. Proprietary software would make the adoption of new hardware more difficult because the development team would have to write custom driver interfaces. This option would require more work to attain a capability equal to the current autonomous operating systems available.

This option includes a great deal of redundant work that would not advance the state of the art. This option is not recommended.

Option 3 would strip out the ANCÆUS control station software, keeping the low-level ANCÆUS protocol for vehicle controllers only. Communication to ANCÆUS hardware would be done through an interpreter that would translate the commands from the new/existing software technology to the low-level ANCÆUS compatible hardware. This option be more flexible than option 1 because ALS would not be limited to current ANCÆUS hardware. It would make the adoption of new hardware more difficult because we would need to adapt ANCÆUS driver interface software ourselves.

Option 4 would adopt a new/existing software technology for use on all ALS systems but would not be limited the use of ANCÆUS protocol. The new/existing software technology would be a candidate open source autonomous operating system like Player/Stage or CARMEN. This option would support but would *not* standardize on ANCÆUS hardware and software. An interpreter would be included to communicate with all legacy ANCÆUS platforms. The advantage of this option is that it allows the use of all legacy hardware and robot platforms, though any adopted operating system will not align completely with the objective of ANCÆUS support.

Option 5 would adopt a new/existing software technology and not adopt any ANCÆUS software and hardware technology for future ALS systems. This option would choose a current autonomous robotic operating system as the baseline for current and future work. The advantage of this option is that we could use a pre-existing software base without modification. One disadvantage of this option is that it does not immediately allow the use of legacy ANCÆUS hardware and software on future ALS programs.

Option 6 would ignore all previous work, designing and developing an autonomous operating system for software and hardware from the ground up. One advantage of this option is that the design could meet all the design criteria that the ALS program deems important. The disadvantage to this option is that to advance this operating system to an acceptable level, significant and largely redundant effort would be required over many years to match the capability of current commercial/open source autonomous systems. This option is not recommended.

These options represent a sliding scale of commitment to ANCÆUS technology. The consensus of the team lies between option 3 and 5. In general the group feels the ANCÆUS technology was a remarkable achievement by a small staff using mid-90's technical standards. The group feels that there is no reason ANCÆUS support could not be provided by future ALS work and strongly recommends this should be done. However, the group felt that for a variety of technical reasons, the adoption of ANCÆUS without modification (i.e. option 1) would not be acceptable and that a major overhaul (option 2) would be costly and redundant with other commercial and open source systems. Nevertheless, the ANCÆUS project has, over time, developed a consistent and useful command set that could easily be adopted for future ALS equipment. At the very least, the team recommends supporting ANCÆUS through option 4, perhaps using option 3 if the API proves sufficiently flexible.

References

1. Cdl systems ltd. brochure - vehicle control station (vcs), 2002. http://www.cdlsystems.com/DocsinPDF/CDL_VCSbrochure.pdf.

2. Cdl systems - vehicle control station (vcs), Novermber 4 2003. http://www.cdlsystems.com/vcs.html.

3. CDL Systems, Calgary, Canada. *Advanced Development Model Control Station Software Configuration*, November 16 1998. For the Improved Landmine Detection Project ILDP-521.

4. CDL Systems, Calgary, Canada. *CCMAT Software Design Document*, March 31 2003.

5. CDL Systems, Calgary, Canada. *CCMAT VCS Enhancements*, April 2 2003. Final Report.

6. Alberta Research Council. *Establishment of a Technology Transfer Agency for the ANCÆUS Vehicle Control System : Final Report*. Alberta Research Council, January 30, 1996.

7. Ron Eirich and Alan Kramer. *ANCAEUS Generic Vehicle Control System Version 2.1.0 (22/01/01)*. Defence Research Establishment Suffield, January 22, 2001.

8. Estlin T. Volpe R. Nesna I. Mutz D. Fisher F. Decision-making in a robotic architecture for autonomy. 2001.

9. Volpe R. Nesnas I. Estlin T. Mutz D. Petras R. Das H. Claraty: Coupled layer architecture for robotic autonomy. *NASA Technical Report*, 2000.

10. Volpe R. Nesnas I. Estlin T. Mutz D. Petras R. Das H. The claraty architecture for robotic autonomy. *Proceedings of the 2001 IEEE Aerospace Conference*, 2001.

11. John Holotko Jr. What is qt?, March 12 2000. http://thor.prohosting.com/~nupshot21/qt_main/node2.shtml.

12. Guzzioni D. Cheyer A. Julia A. Konolige K. Many robots make short work. *Proceedings of the AAAI*, 1996.

13. Kurt Konolige and Karen Myers. The saphira architecture for autonomous mobile robotics. *SRI Report*, 1996.

14. Reid Simmons and Dale James. *A Reference Manual for IPC version 3.4*. Carnegie Mellon University, February, 2001.

15. Nesnas I. Wright A. Bajrachayra M. Simmons R. Estlin T. Kim W. Claraty: An architecture for reusable robotic software. *SPIE Aerosense Proceedings 2003*, 2003.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence R & D Canada - Suffield
PO Box 4000, Medicine Hat, AB, Canada T1A 8K6

2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable).

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

Engineering Review of ANCÆUS/AVATAR

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Verret, G. Broten, D. Erickson, J. Giesbrecht, S. Monckton, S.

5. DATE OF PUBLICATION (month and year of publication of document)	6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc).	6b. NO. OF REFS (total cited in document)
December 2003	46	15

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

Technical Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include address).

Defence R & D Canada - Suffield
PO Box 4000, Medicine Hat, AB, Canada T1A 8K6

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Specify whether project or grant).	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written).

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique.)	10b. OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)
DRDC Suffield TR 2003-167	

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Defence departments and defence contractors; further distribution only as approved
( ) Defence departments and Canadian defence contractors; further distribution only as approved
( ) Government departments and agencies; further distribution only as approved
( ) Defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution beyond the audience specified in (11) is possible, a wider announcement audience may be selected).

Unlimited

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

Following the guidelines of the Engineering Review Terms of Reference, this report examines whether ANCÆUS and/or VCS constitute an enabling technology for future Autonomous Land System (ALS) autonomous vehicles. Since 1989, forward thinking engineering has characterized the history of ANCÆUS and VCS. ANCÆUS has broken ground in network vehicle architectures, vehicle modularity, hardware self-discovery and robust multi-vehicle networking. Similarly, VCS has grown into a major business for CDL Systems Inc. through consistent technical superiority. Though conceived over 15 years ago, these systems remain sufficiently relevant and desireable to serve within the ILDS deployed to Afghanistan in 2003. This report reviews the historical background and current technical content of ANCÆUS, VCS, ALS and similar commercial/open source efforts. The subsequent comparative evaluation of ANCÆUS and commercial/open source technologies revealed that currently no single system constitutes a complete a multi-robot autonomous system architecture. The report concluded that while ANCÆUS was found to be technically well concieved, and often well executed, commercial/open source technologies are more capable in a multi-vehicle context and represent a lower technical risk. Specifically, the report concludes:

1. ANCÆUS is not an enabling technology for future autonomous vehicles and that
2. ALS should adopt commercial/open source technology to support a new ALS architecture and
3. ALS should provide ANCÆUS compatibility and evaluate the ANCAEUS control codes as a guideline for future ALS vehicle interfaces.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

ANCAEUS
VCS
Teleoperation
Remote Control
Robot
Autonomous Vehicle